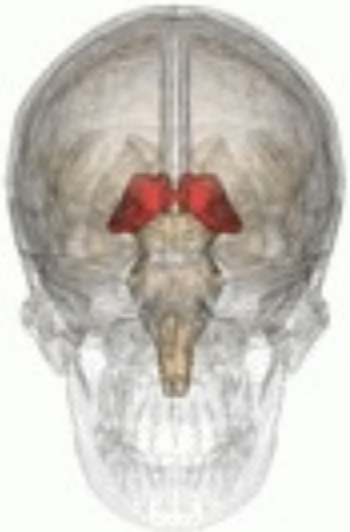
EPFL Blue Brain Project

Soplata Job Talk, 2024-05-09, For HNN Software Engineer Position

Download this talk here:
asoplata.com/talk

Austin Soplata, PhD
Postdoc for Thalamus Special Region, Circuits Team

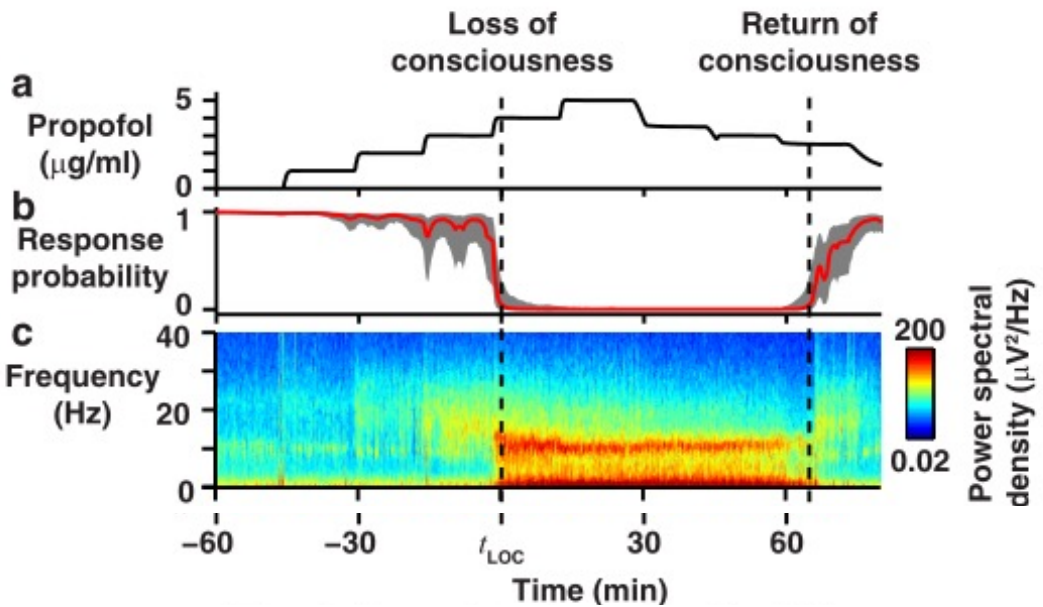
The thalamus: If you're reading this, you have one (probably)



(not to scale)

Ref: Wikipedia, "Thalamus", "Walnut"

PhD and first postdoc: Understanding propofol anesthesia via modeling EEG oscillations

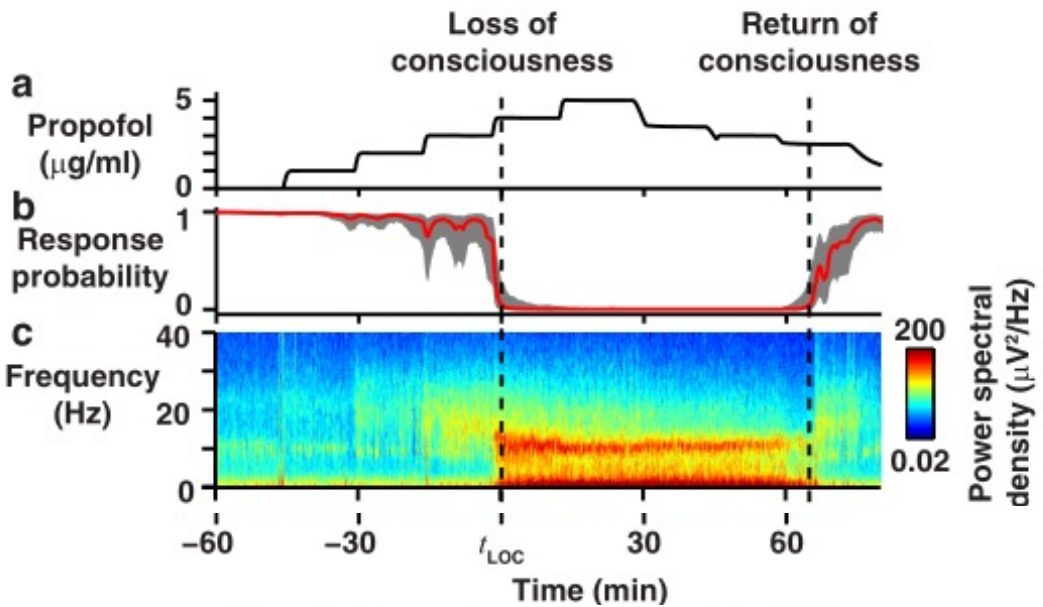


- What causes Alpha Oscillations (8-14 Hz) in propofol anesthesia?
- What causes Slow Wave Oscillations (SWO, 0.1-2 Hz) in propofol anesthesia?
- What causes the change in **Phase-Amplitude Coupling** between alpha and SWO?

EPFL

Ref: Mukamel, et al., 2014 https://doi.org/10.1523%2FJNEUROSCI.5813-12.2014

PhD and first postdoc: Understanding propofol anesthesia via modeling EEG oscillations

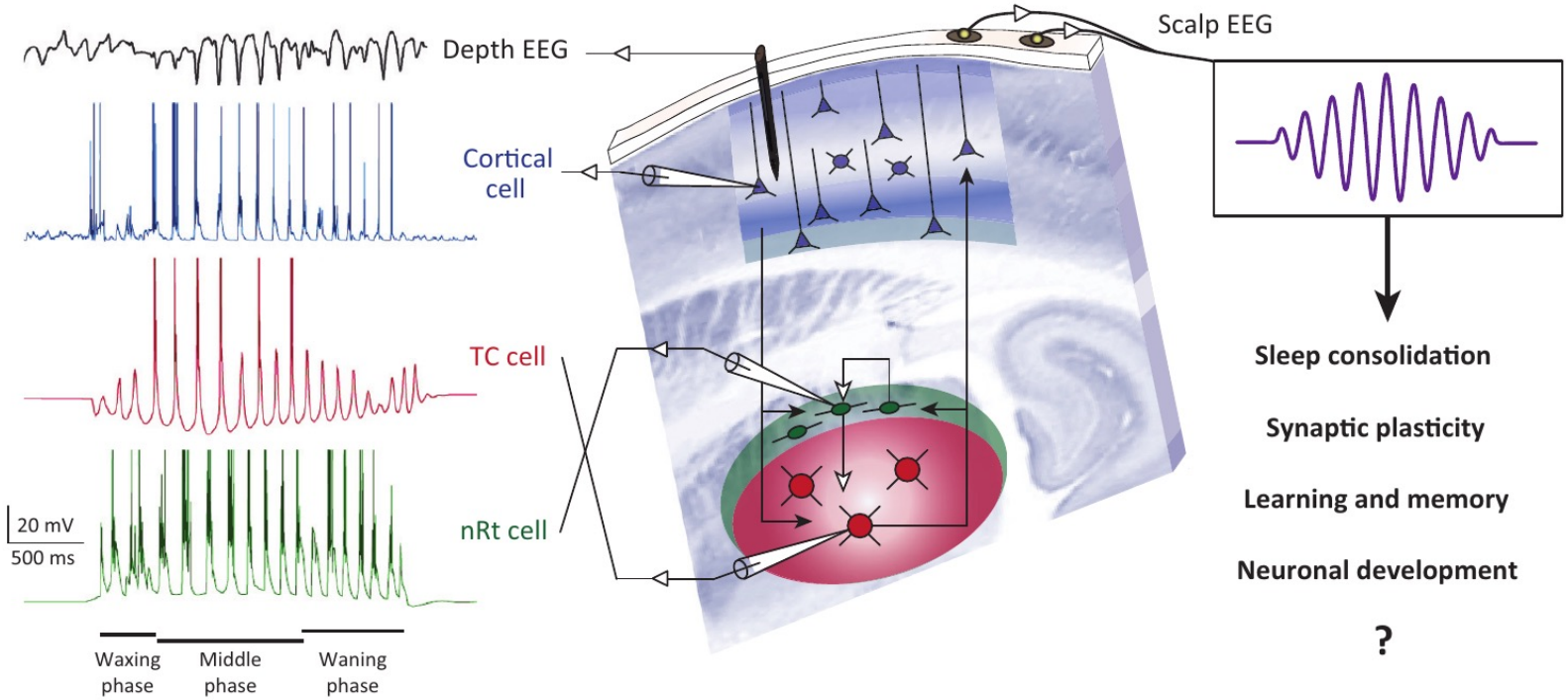


- What causes Alpha Oscillations (8-14 Hz) in propofol anesthesia?
- What causes Slow Wave Oscillations (SWO, 0.1-2 Hz) in propofol anesthesia?
- What causes the change in **Phase-Amplitude Coupling** between alpha and SWO?

EPFL

Ref: Mukamel, et al., 2014 https://doi.org/10.1523%2FJNEUROSCI.5813-12.2014

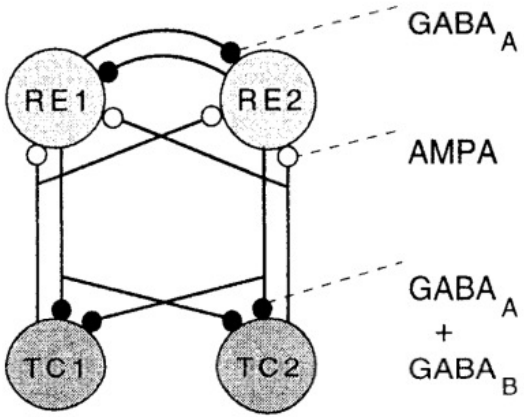
Propofol alpha is same frequency as Thalamic Sleep Spindles (8-16 Hz)

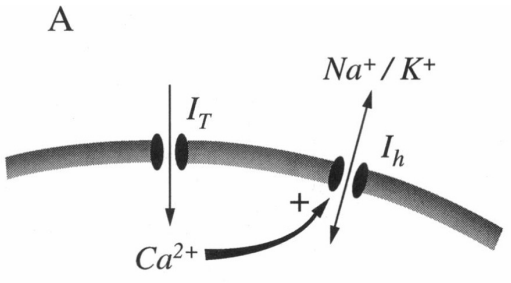


EPFL

Ref: Astori, et al., 2013 https://doi.org/10.1016/j.tins.2013.10.001

Destexhe Model of Thalamic Spindles



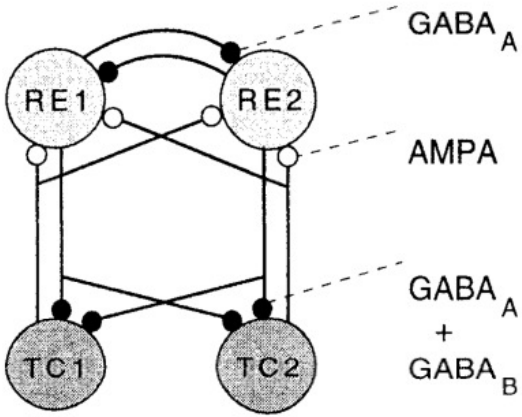


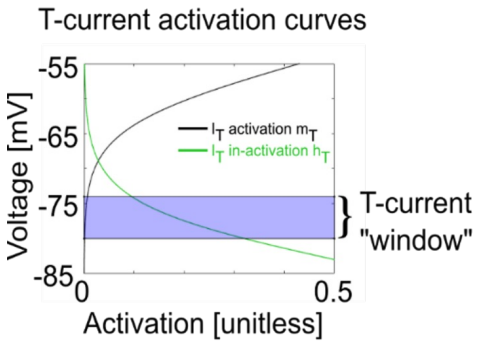
$$C_m \dot{V}_T = -g_L(V_T - E_L) - I_T - I_h - I_{KL} - I_{Na} - I_K - I_{GABA_A T} - I_{GABA_B} \quad (1)$$

EPFL

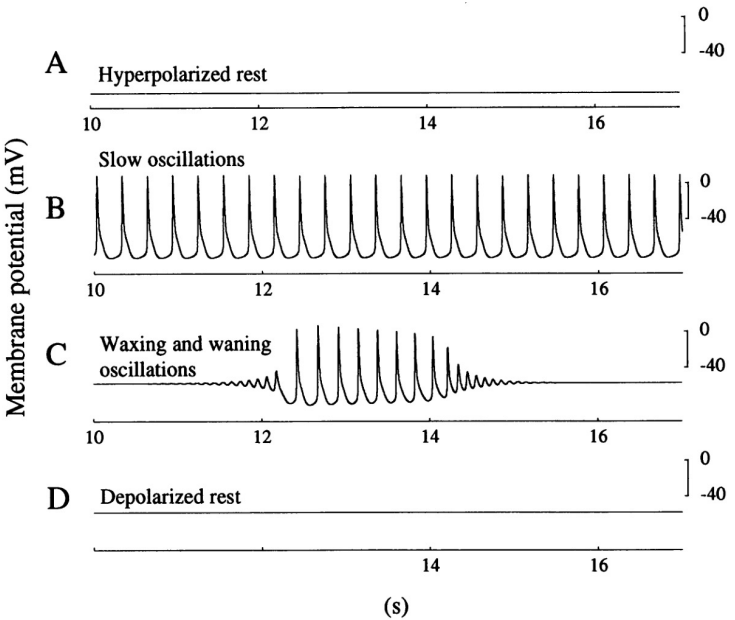
Ref: Destexhe, et al., 1996 https://doi.org/10.1152/jn.1996.76.3.2049 and 1993

Destexhe Model of Thalamic Spindles





Different H-current simulations



$$C_m\dot{V}_T = -g_L(V_T - E_L) - I_T - I_h - I_{KL} - I_{Na} - I_K - I_{GABA_AT} - I_{GABA_B} \quad (1)$$

Ref: Destexhe, et al., 1996 https://doi.org/10.1152/jn.1996.76.3.2049 and 1993

frontiers in Neuroinformatics

METHODS
published: 15 March 2018
doi: 10.3389/fninf.2018.00010

DynaSim: A MATLAB Toolbox for Neural Modeling and Simulation

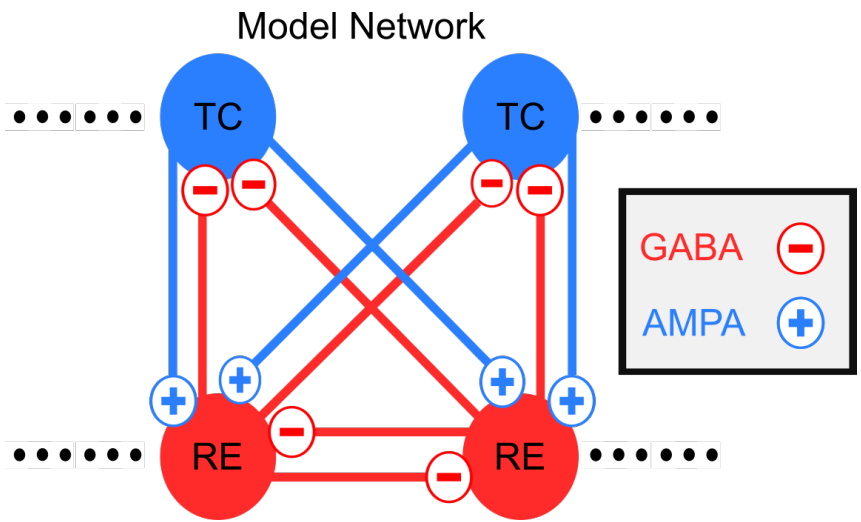
Jason S. Sherfey[1,2], Austin E. Soplata[3], Salva Ardid[1], Erik A. Roberts[4], David A. Stanley[1], Benjamin R. Pittman-Polletta[1] and Nancy J. Kopell[1]*

- Easy vectorization of ODEs
- Plug-and-play mechanism functionality like NEURON MOD files
- Built-in parameter grid search and batch job submission on clusters/HPC
- …however, not actively developed for the last 2-3 years

EPFL

Ref: Sherfey, et al., 2018 https://doi.org/10.3389/fninf.2018.00010

First Thalamic Circuit



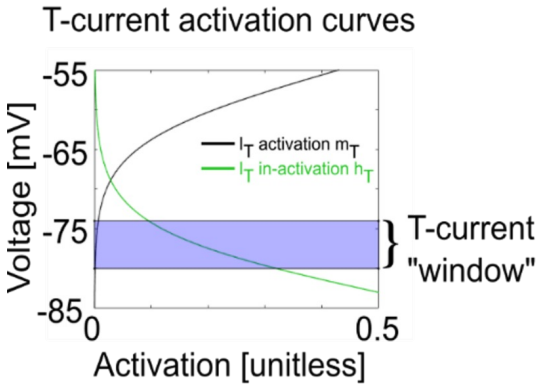
Propofol direct effects

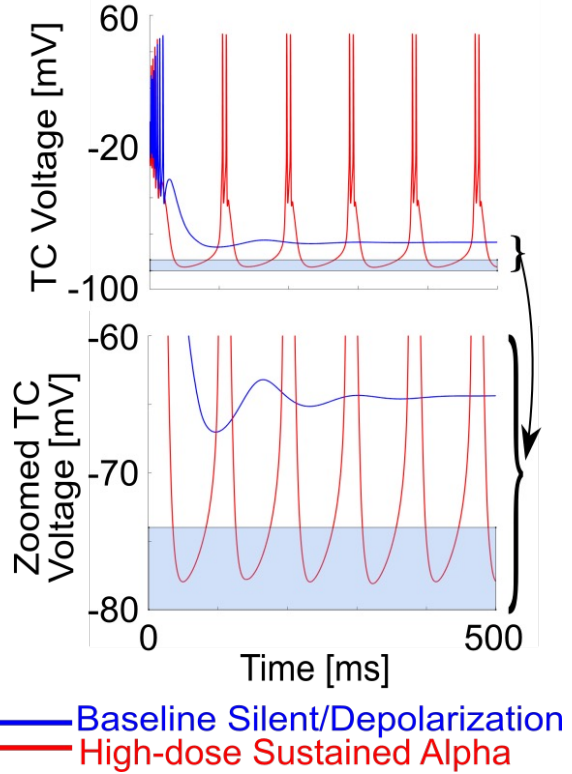
- Increases $\bar{g}_{GABA_A}$
  ("strength of inhibition")
- Increases τ_{GABA_A}
  ("how long inhibition lasts")
- Decreases $\bar{g}_H$
  (TC cell H-current strength)
- Decreases Background Excitation

EPFL

Ref: Soplata, et al., 2017 https://doi.org/10.1371/journal.pcbi.1005879 unless otherwise indicated

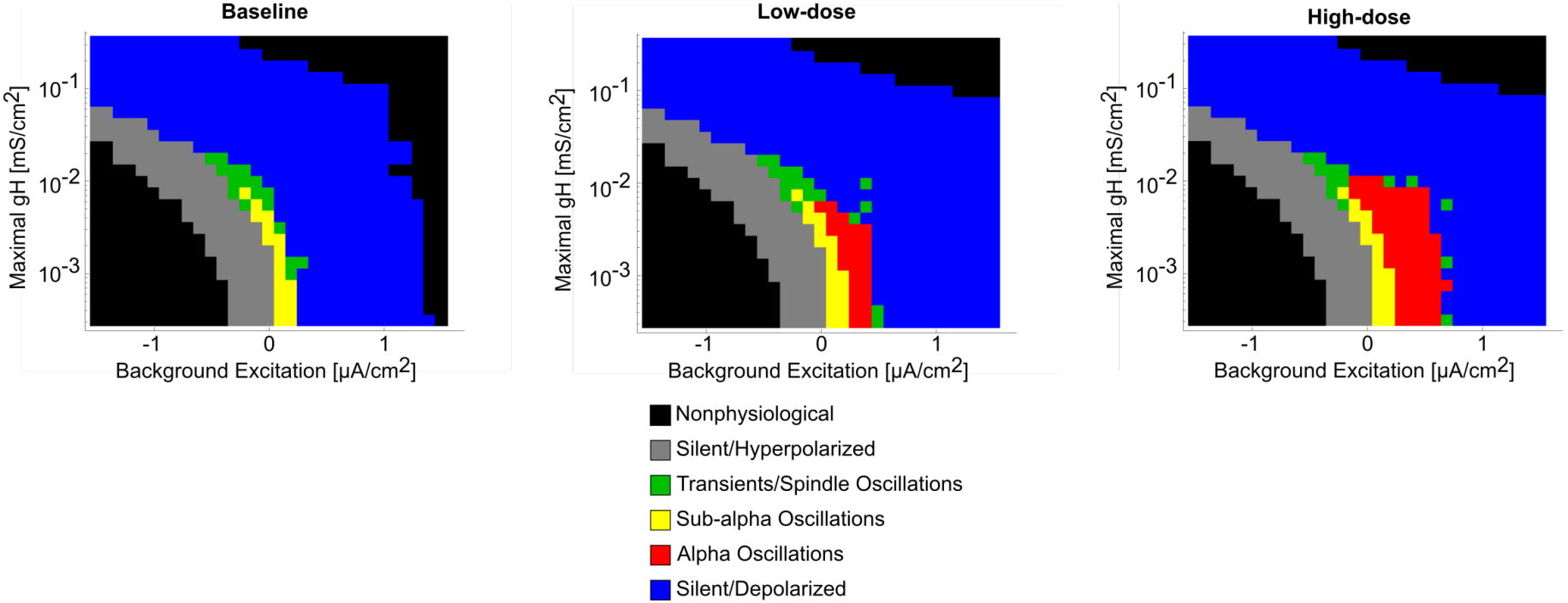
Enhanced $GABA_A$ inhibition enables Alpha





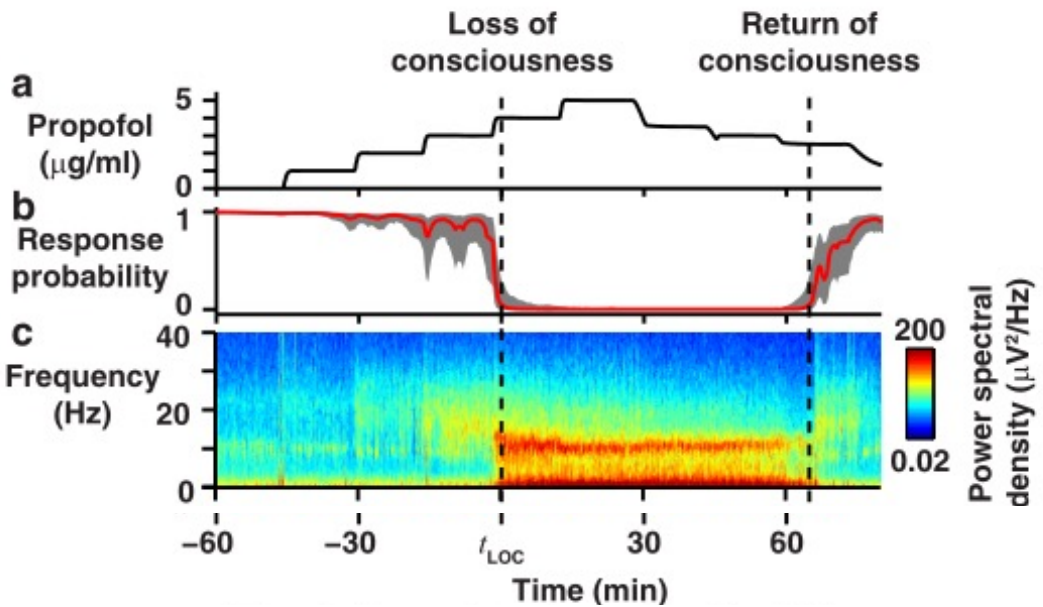
EPFL

Propofol changes to **$GABA_A$ and H-current** affect the likelihood of Alpha



EPFL

Understanding propofol anesthesia via modeling EEG oscillations

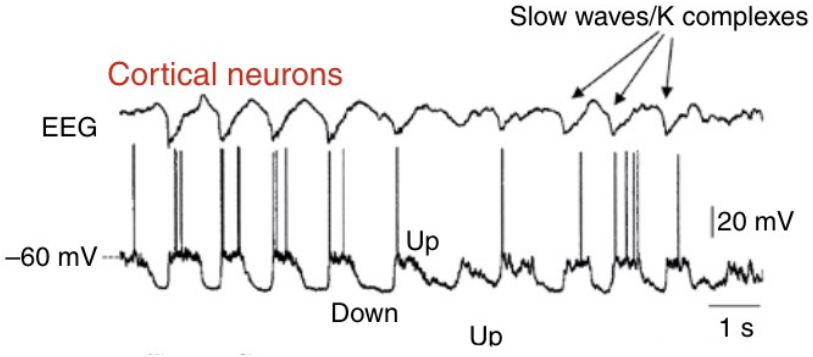


- What causes Alpha Oscillations (8-14 Hz) in propofol anesthesia?
- What causes Slow Wave Oscillations (SWO, 0.1-2 Hz) in propofol anesthesia?
- What causes the change in **Phase-Amplitude Coupling** between alpha and SWO?

EPFL

Ref: Mukamel, et al., 2014 https://doi.org/10.1523%2FJNEUROSCI.5813-12.2014

Sleep Slow-Wave Oscillations (SWO) (~1 Hz)

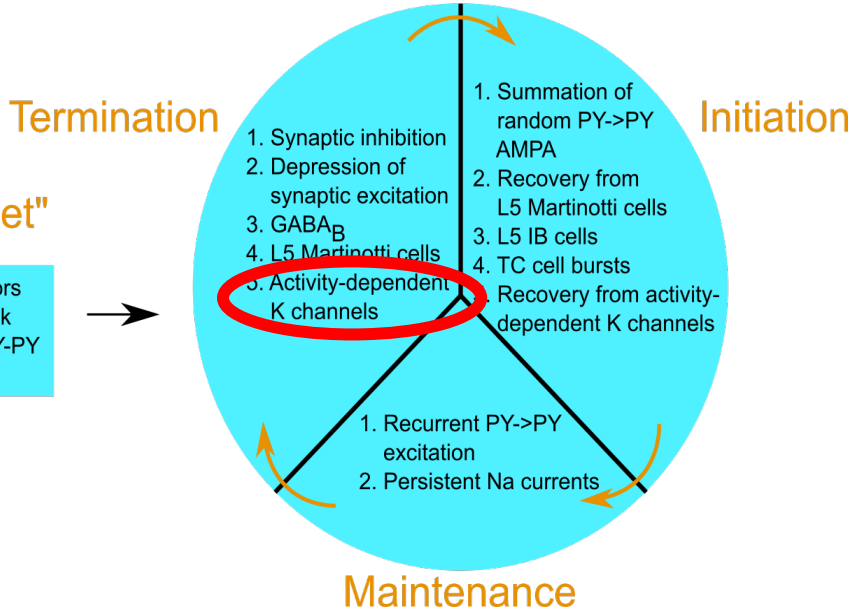


SWO "onset"

1. Neuromodulators affecting K-Leak
2. Decrease in PY-PY excitation

→

Slow Wave Oscillation Mechanisms



Ref: Crunelli & Hughes, 2010 https://doi.org/10.1038/nn.2445 and the Internet

Second, Thalamo-cortical circuit

Cortical Slow Wave Mechanism:
K(Na)-current

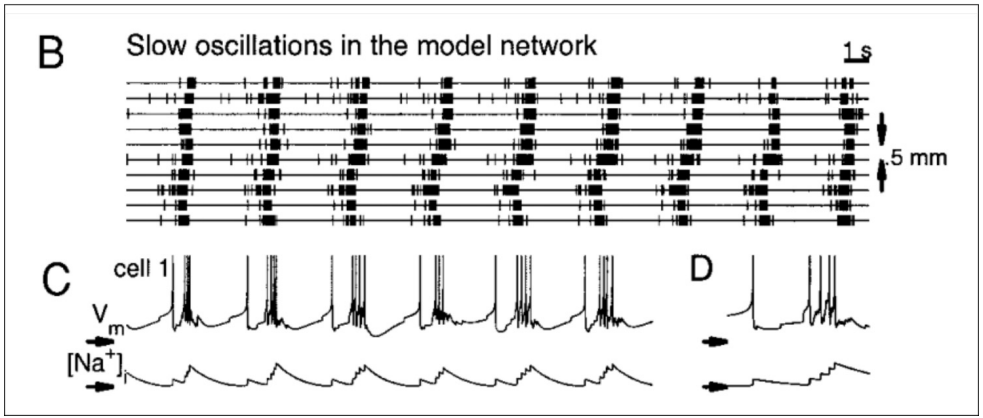


Image from (Compte et al., 2003)

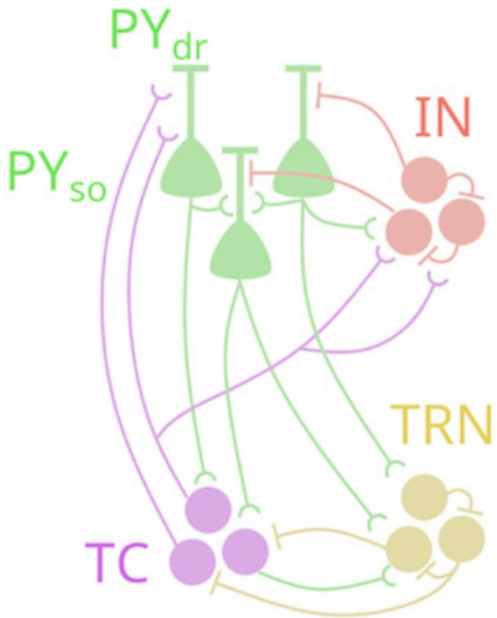
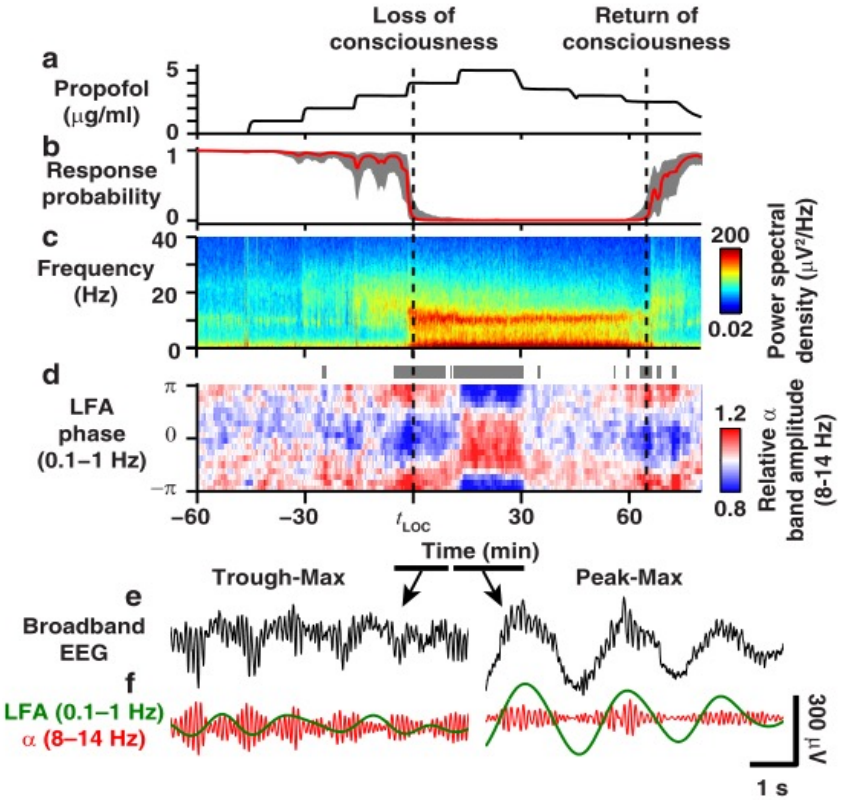


Image from (Soplata et al., 2023)

EPFL

Ref: Soplata et al., 2023 https://doi.org/10.1152/jn.00068.2022

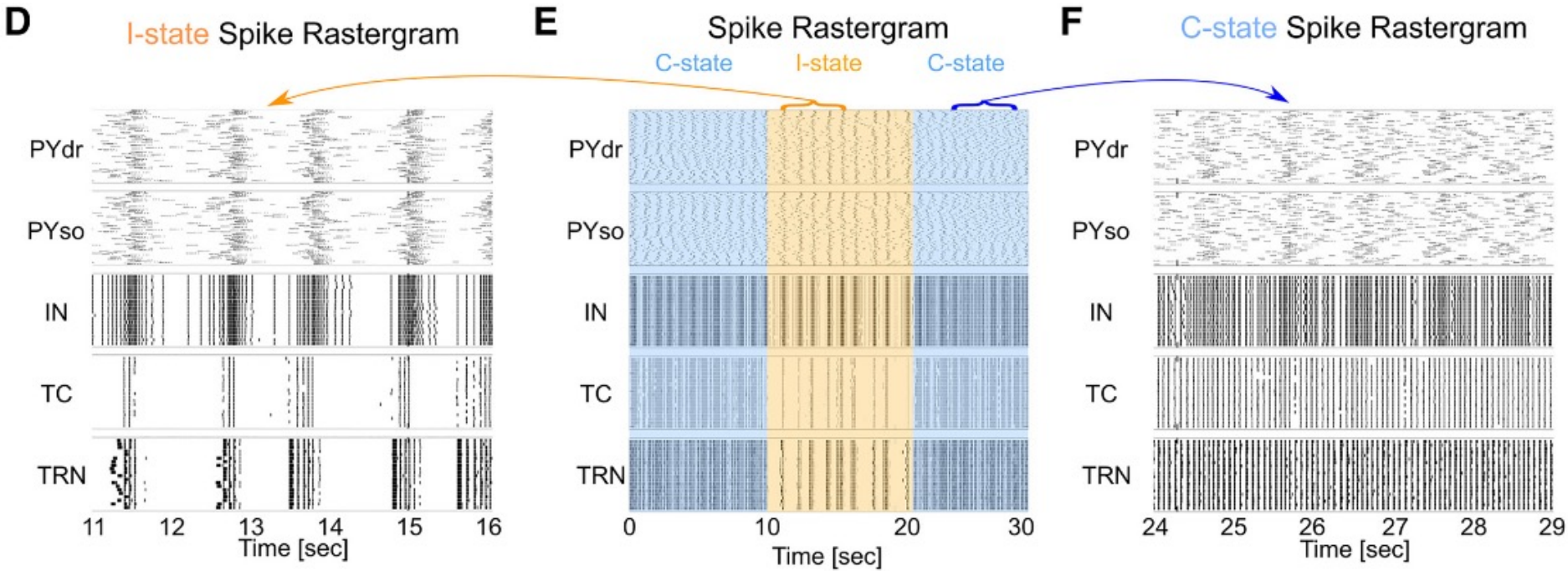
Understanding propofol anesthesia via modeling EEG oscillations



- What causes Alpha Oscillations (8-14 Hz) in propofol anesthesia?
- What causes Slow Wave Oscillations (SWO, 0.1-2 Hz) in propofol anesthesia?
- What causes the change in **Phase-Amplitude Coupling** between alpha and SWO?

Ref: Mukamel, et al., 2014 https://doi.org/10.1523%2FJNEUROSCI.5813-12.2014

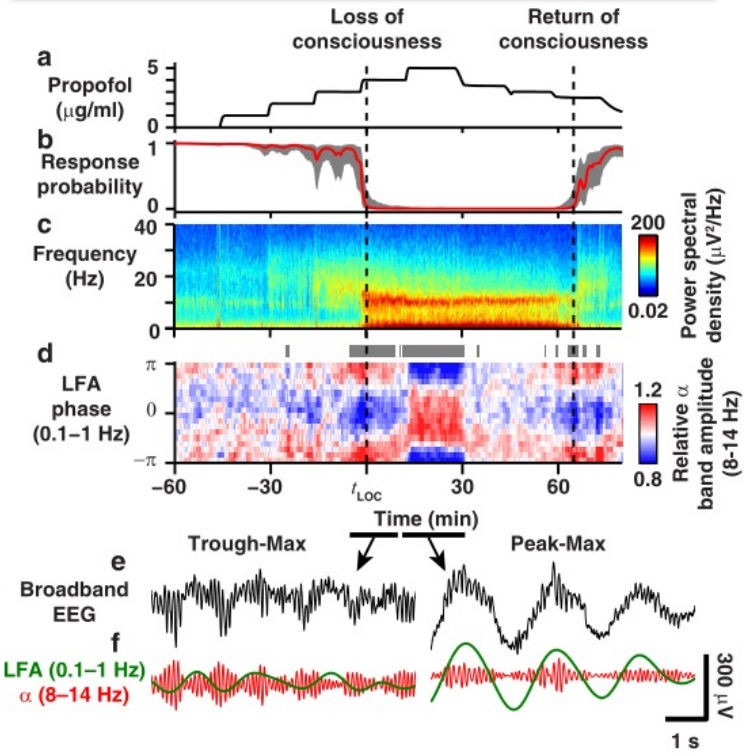
Our thalamocortical circuit would randomly switch between two oscillatory states



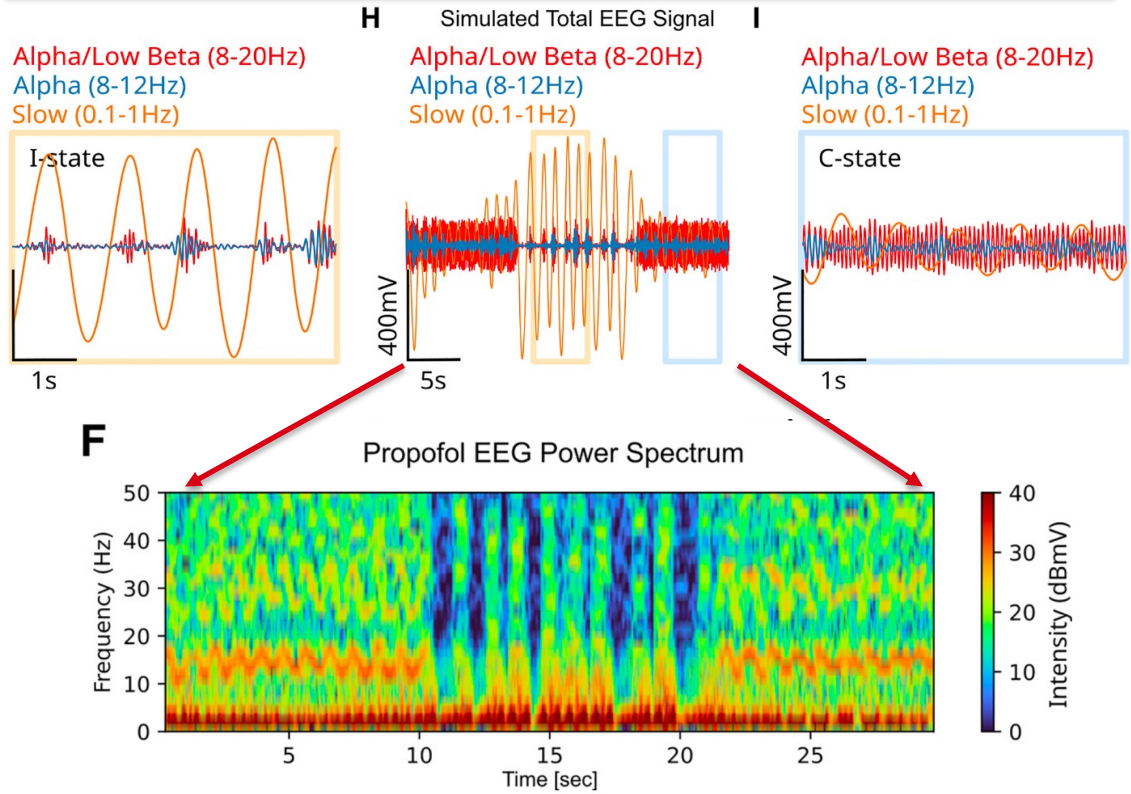
Ref: Soplata et al., 2023 https://doi.org/10.1152/jn.00068.2022

The two simulated states were phase-amplitude coupled like propofol EEG data

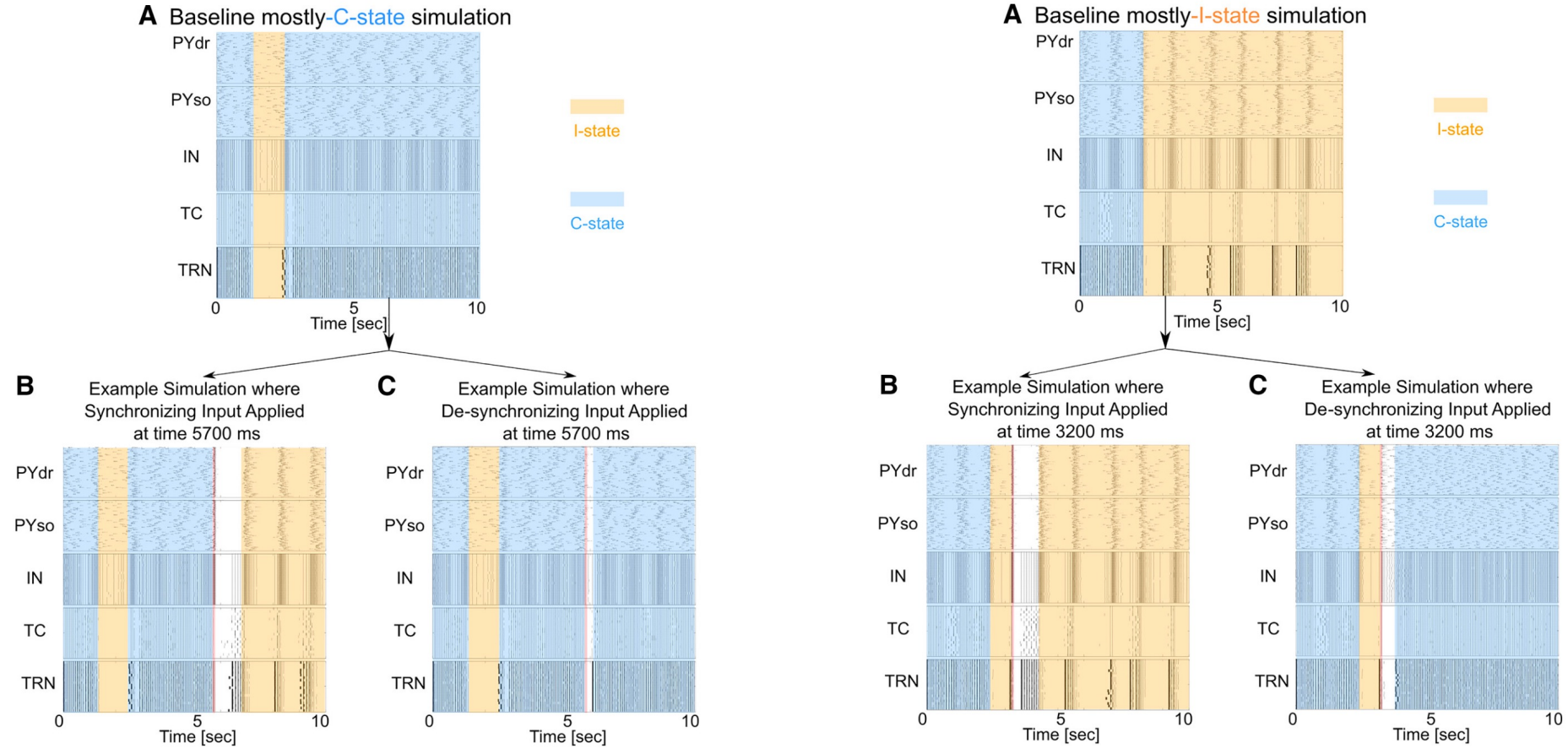
Experiment



Simulation

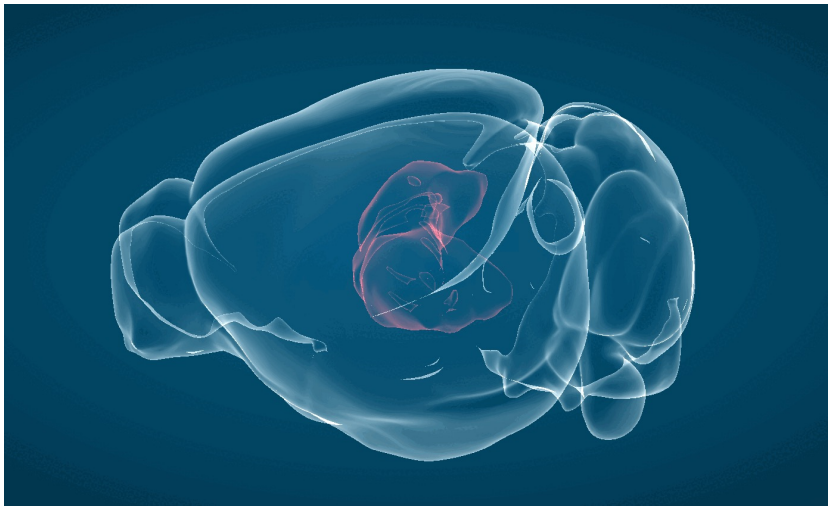


The thalamocortex switched coupling type based on **dynamic cortical synchronization** level



Second postdoc: Blue Brain Project

- Here since beginning of 2023
- 90% of my time has been learning how to use, and quickly get up to speed with, many internal and some external Python libraries
- (Yes, this has left very little time for “doing science”…)
- Most of what I will show is unpublished and internal, but will be open-sourced at end of 2024. (If you want, I can send you a copy now, just ask)



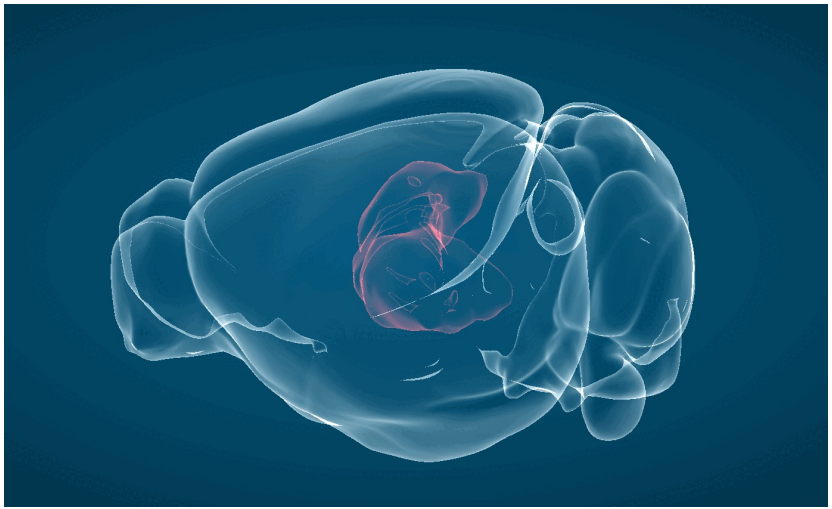
Ref: Allen CCFv3 mouse brain, thalamus highlighted, all following images Unpublished unless otherwise indicated

NEURON Caveats

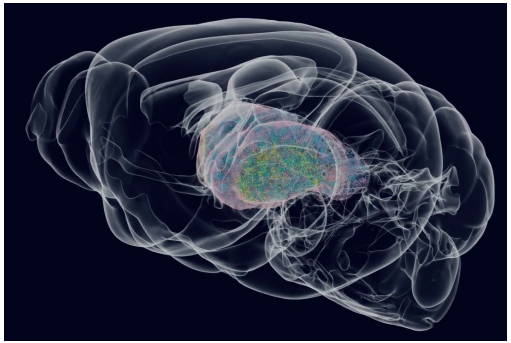
- To be clear, 99% of “NEURON programming” is handled by other teams:
  - **Cells Team:** Scientists who fit single-cell models based on reconstructed cell morphologies and electrophysiological data. They provide parametrized celltypes for our models.
  - **Neuroscientific Software Engineering:** SoftWare Engineers (SWEs) who do general software management and maintenance, including "production-izing" scientific code. They support software that "builds" all the different models.
  - **High-Performance Computing**: SWEs specializing in running NEURON at-scale on our supercomputer, including environment management. They support us "running" the models.
  - **Circuits Team** (the one I'm on): Scientists who work with everyone else to "assemble" ***specific*** models (such as for Thalamus) and attempt to "do science" using the models.
  - Many other teams (Data/Knowledge Engineering, Visualization, etc.)
  - Caveat to the caveat: I have used NEURON and models built in it from before BBP

How do we go from mouse brain spatial regions to simulations?

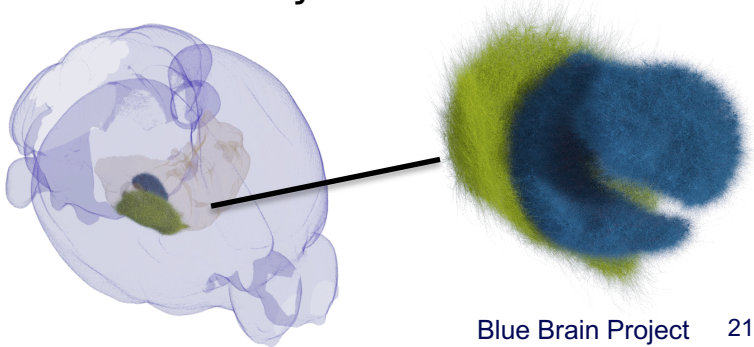
Allen Institute Mouse Brain Atlas
Version CCFv3



Whole thalamus model

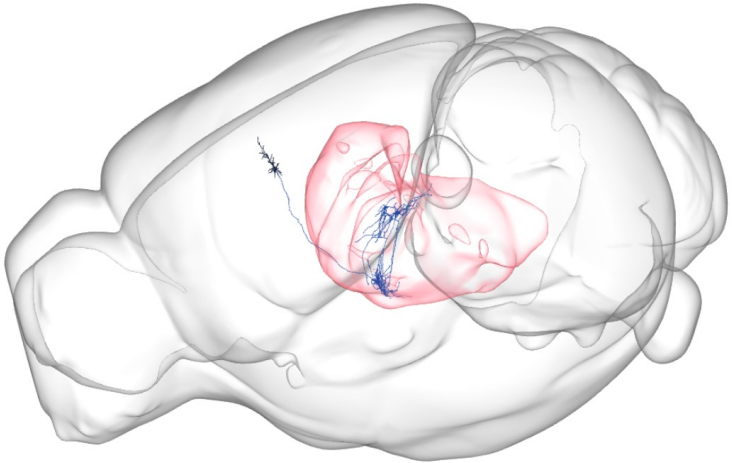


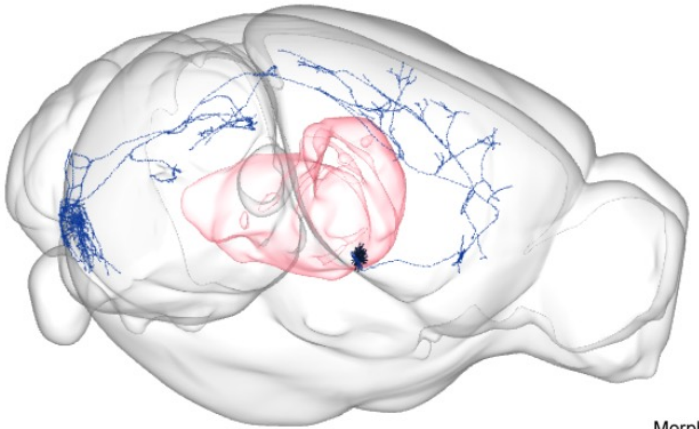
Somatosensory thalamus model



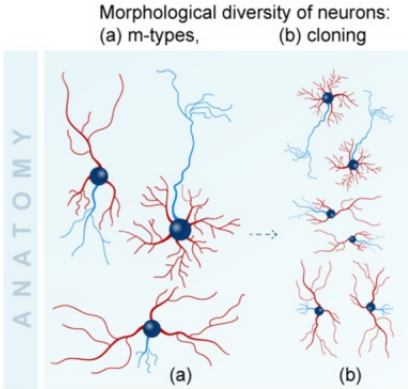
Ref: Allen CCFv3 mouse brain, thalamus highlighted, and unpublished

We begin with cell shapes, or “morphologies”



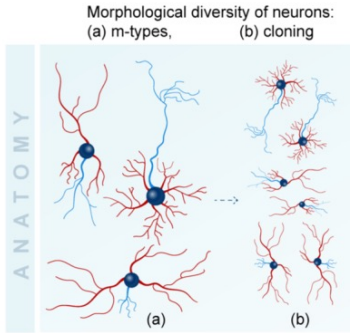


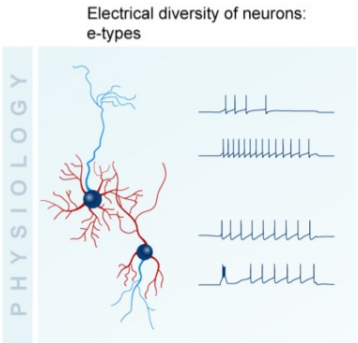
- We start with ”reconstructed” morphologies of real cells provided to us by multiple labs. Only some are provided in “mouse brain space”.
- I have helped to validate whether these morphologies are located in the “correct” spatial regions, including between different coordinate systems (Allen’s CCFv3 vs Janelia’s “legacy CCFv2.5”).

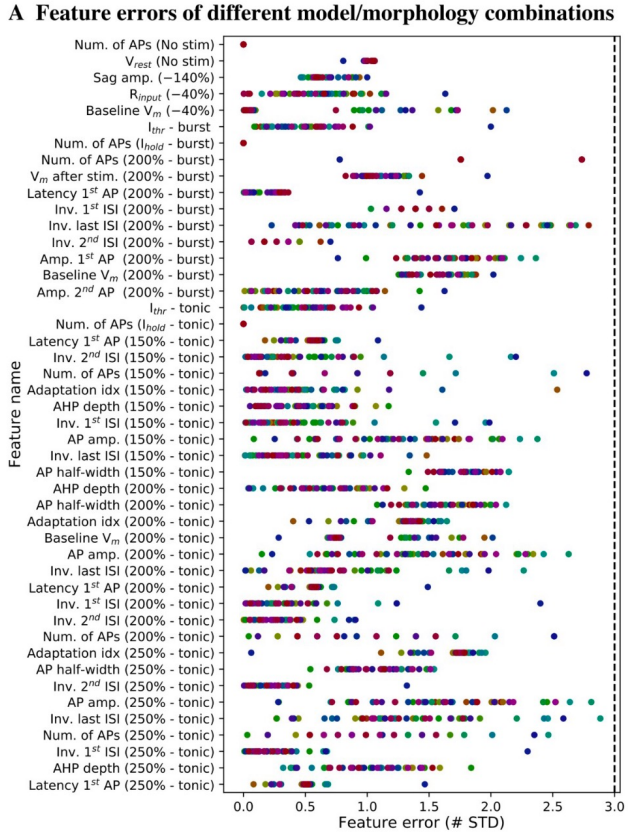


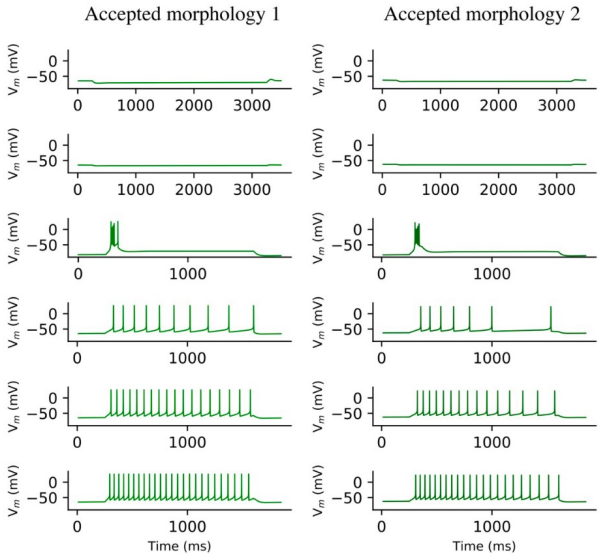
Refs: Janelia Mouselight, AA0138 and AA0799 and (Iavarone, et al., 2019 https://doi.org/10.1371/journal.pcbi.1006753)

We then create MANY “morpho-electric” celltypes by fitting electrophysiological data to our morphologies (not me though)





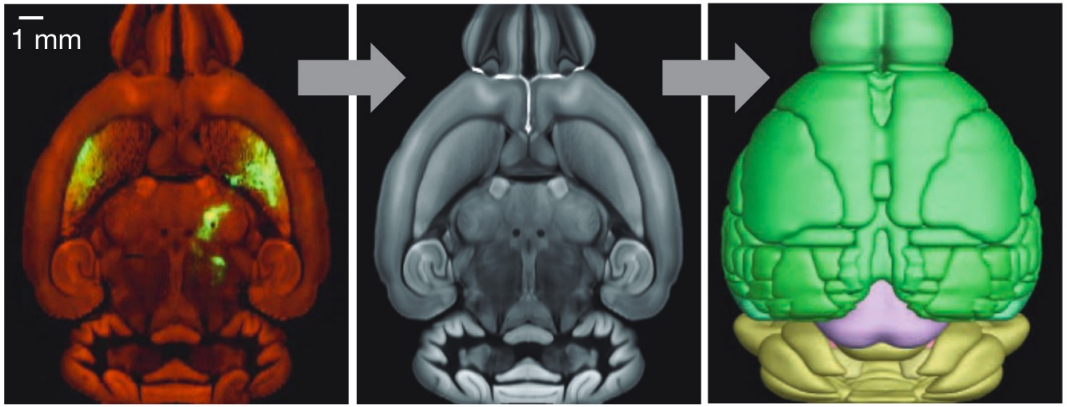




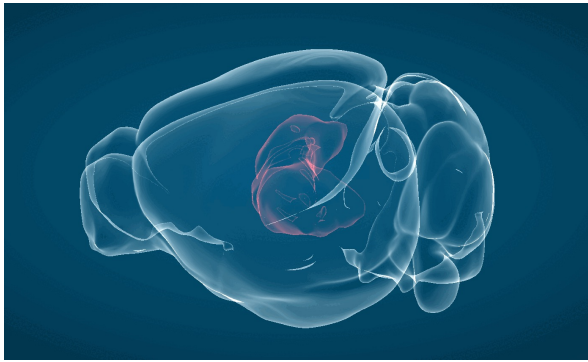
EPFL

Refs (Iavarone, et al., 2019 https://doi.org/10.1371/journal.pcbi.1006753)

Where do we put our cells? Regions: The Allen Mouse Brain Atlas



Single specimen | Average template brain | 3D reference model



Consists of:

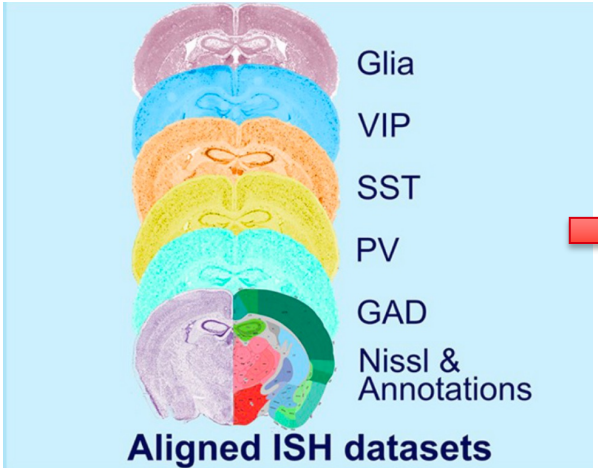
1. A 3-dimensional NRRD “annotation” file that assigns every 25um^3 voxel to a single region ID
2. A “hierarchy” JSON file that identifies (almost) every region in the brain

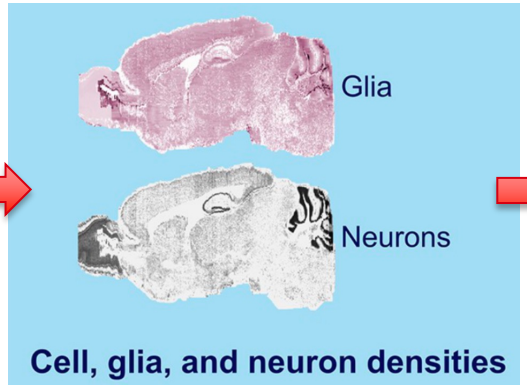

```
{
 "id": 733,
 "atlas_id": 374,
 "ontology_id": 1,
 "acronym": "VPM",
 "name": "Ventral posteromedial nucleus of the thalamus",
 "color_hex_triplet": "FF8084",
 "graph_order": 649,
 "st_level": 9,
 "hemisphere_id": 3,
 "parent_structure_id": 709,
 "children": []
},
```

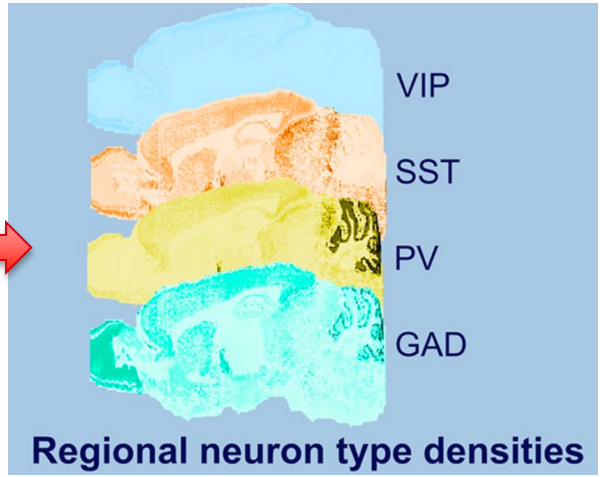

EPFL

Ref: (Oh, et al., 2014 https://www.nature.com/doifinder/10.1038/nature13186)

Where do we put our cells? Cell densities: The BBP Atlas







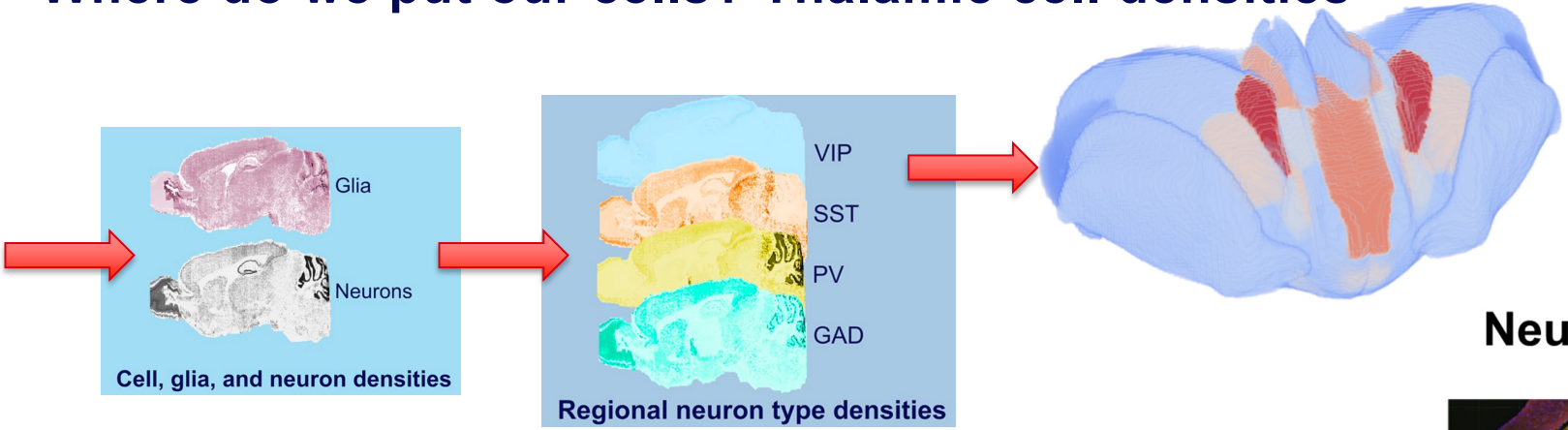
The **BBP Atlas**, which is based on the Allen Atlas, is the result of an extremely complex piece of software, which estimates the "cell density" (cells / mm^3) for most cell types in all brain regions. This consists of:

1. A "hierarchical" JSON file identifying brain regions (including changes from Allen)
2. Hundreds of 3-dimensional NRRD files for densities of different celltypes
3. Other files, including compilations of literature data values

EPFL

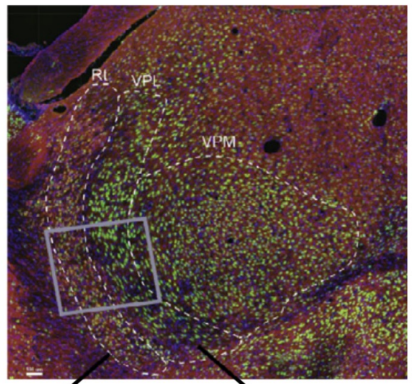
Ref: (Rodarie, et al., 2022 https://doi.org/10.1371/journal.pcbi.1010739)

Where do we put our cells? Thalamic cell densities



As with most things at BBP, the BBP Atlas was only validated for cortex. This required me to do:

- Assistance in creation of Singularity containers to make the BBP Atlas Pipeline reproducible (including a brief foray into Docker)
- Significant debugging of its consumption of our “Nexus” data storage / knowledge base and API
- Validation and correction of our thalamus cell density values at different stages across the Pipeline



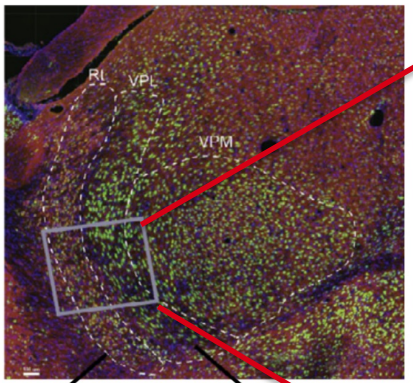
Rt: 68’750 ± 1’976 neurons/µm^3

VPL: 57’467 ± 5’201 neurons/µm^3

Ref: (Rodarie, et al., 2022 https://doi.org/10.1371/journal.pcbi.1010739), (Iavarone, et al., 2023 https://doi.org/10.1016/j.celrep.2023.112200), and unpublished

We've got soma positions, let's go! Example columnar circuit

Neuron density

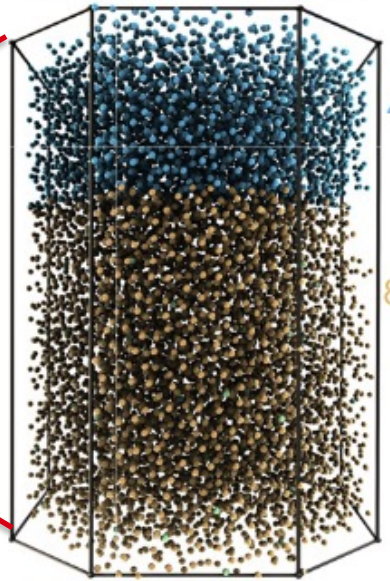


Rt:
68'750 ± 1'976
neurons/µm³

VPL:
57'467 ± 5'201
neurons/µm³

Neuron counts

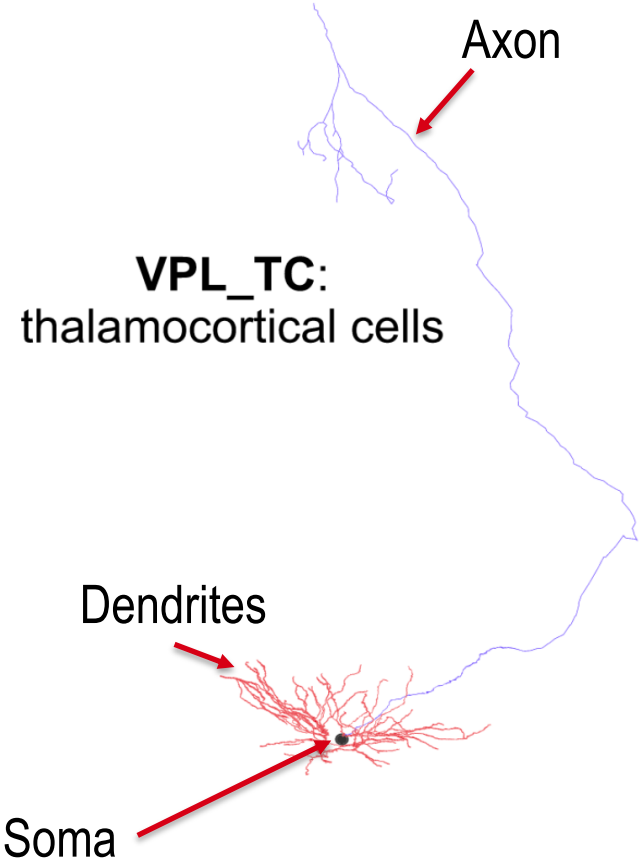
13'908 ± 591 neurons



4'909 ± 283
Rt_RC

8'952 ± 517
VPL_TC

47 ± 2
VPL_IN

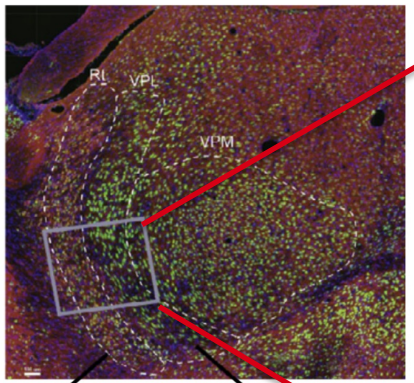


EPFL

Ref: (Iavarone, et al., 2023 https://doi.org/10.1016/j.celrep.2023.112200)

We've got soma positions, let's go! Not so fast...WHERE are we going?

Neuron density

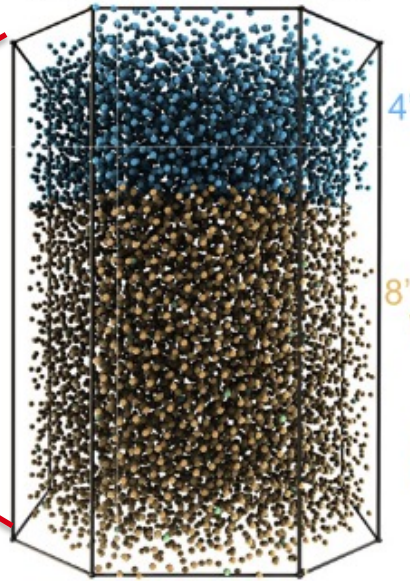


Rt:
68'750 ± 1'976
neurons/µm³

VPL:
57'467 ± 5'201
neurons/µm³

Neuron counts

13'908 ± 591 neurons

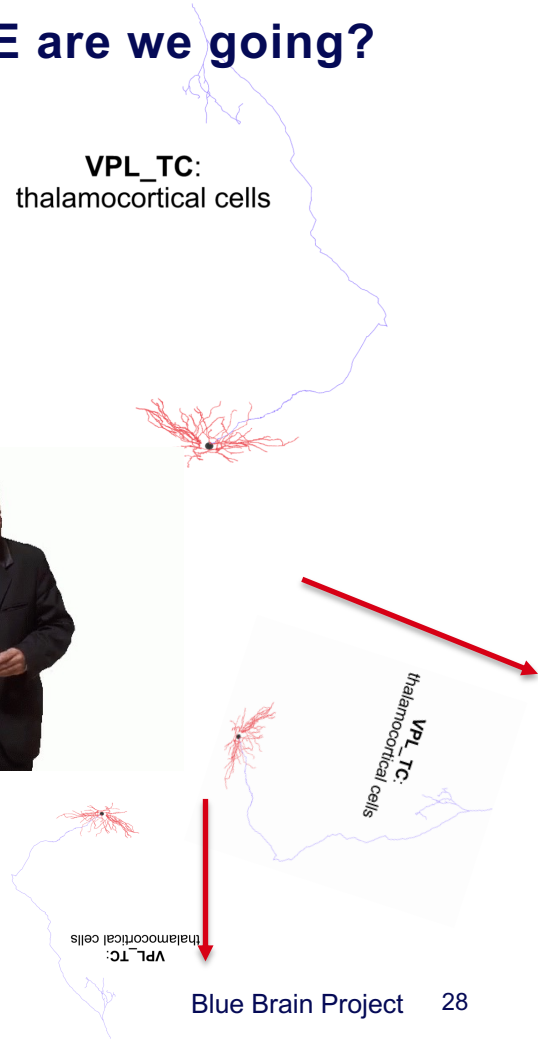


4'909 ± 283
Rt_RC

8'952 ± 517
VPL_TC

47 ± 2
VPL_IN

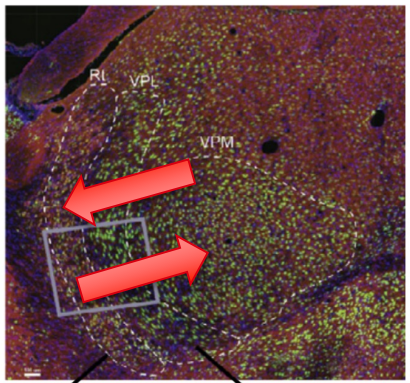
VPL_TC:
thalamocortical cells



Ref: (Iavarone, et al., 2023 https://doi.org/10.1016/j.celrep.2023.112200)

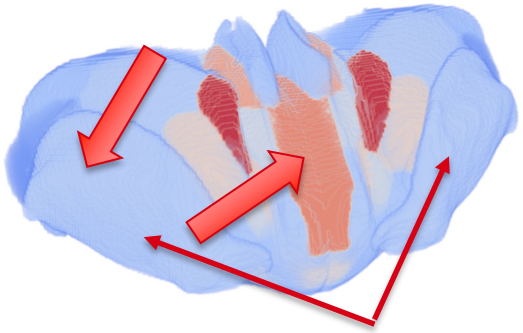
We need to orient the “direction vectors” of our thalamic cells

Neuron density

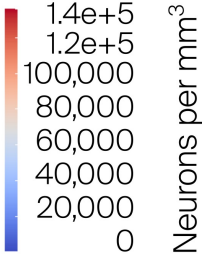


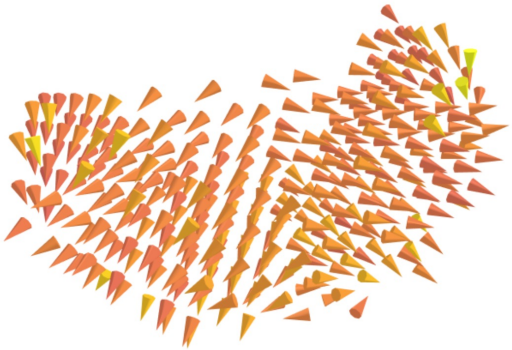
Rt:
68’750 ± 1’976
neurons/µm^3

VPL:
57’467 ± 5’201
neurons/µm^3



RT regions





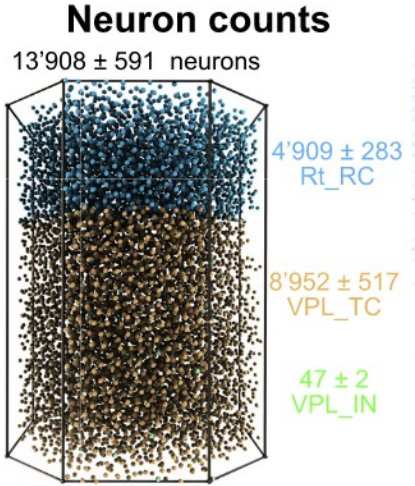
We know from literature that TC cells from inside the thalamus project outwards the RT, and RT cells project in the opposite direction.

A predecessor had made code years ago to create thalamic direction vectors via blurring a gradient, starting from the brain midline and ending at the RT. I updated this code to use our modern toolset, and fixed some major calibration errors.

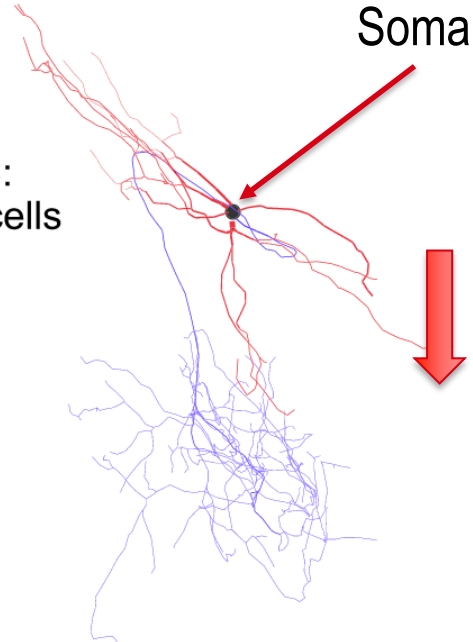
(Note that cells can be defined to go with or ***opposite to*** the vectors.)
This code is public: https://github.com/BlueBrain/atlas-direction-vectors

Ref: (Iavarone, et al., 2023 https://doi.org/10.1016/j.celrep.2023.112200), Unpublished

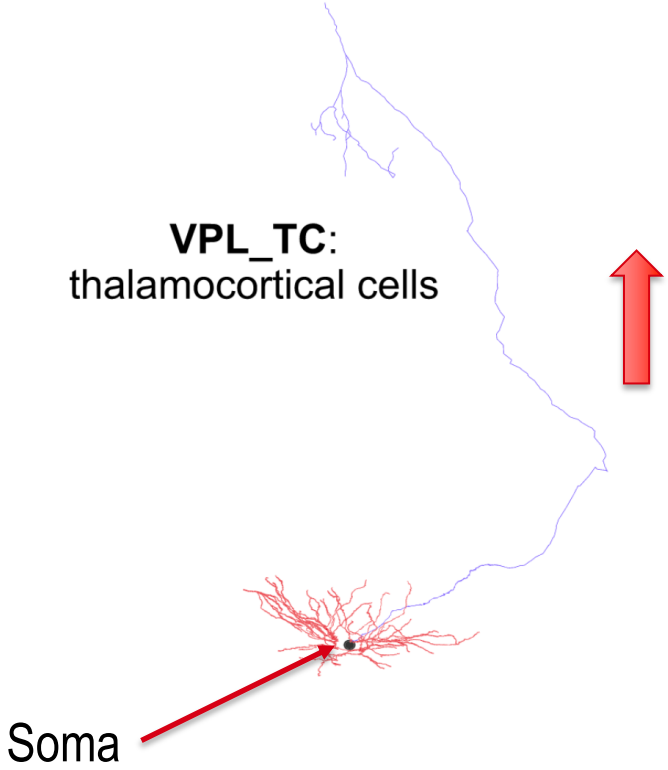
We've got our direction / orientation vectors for every voxel



Rt_RC:
reticular cells

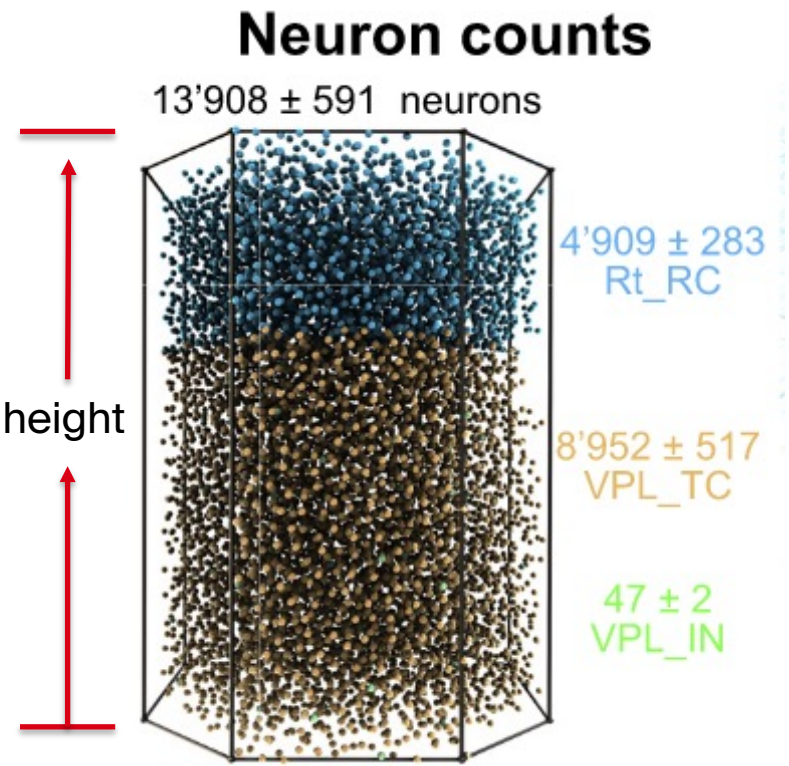


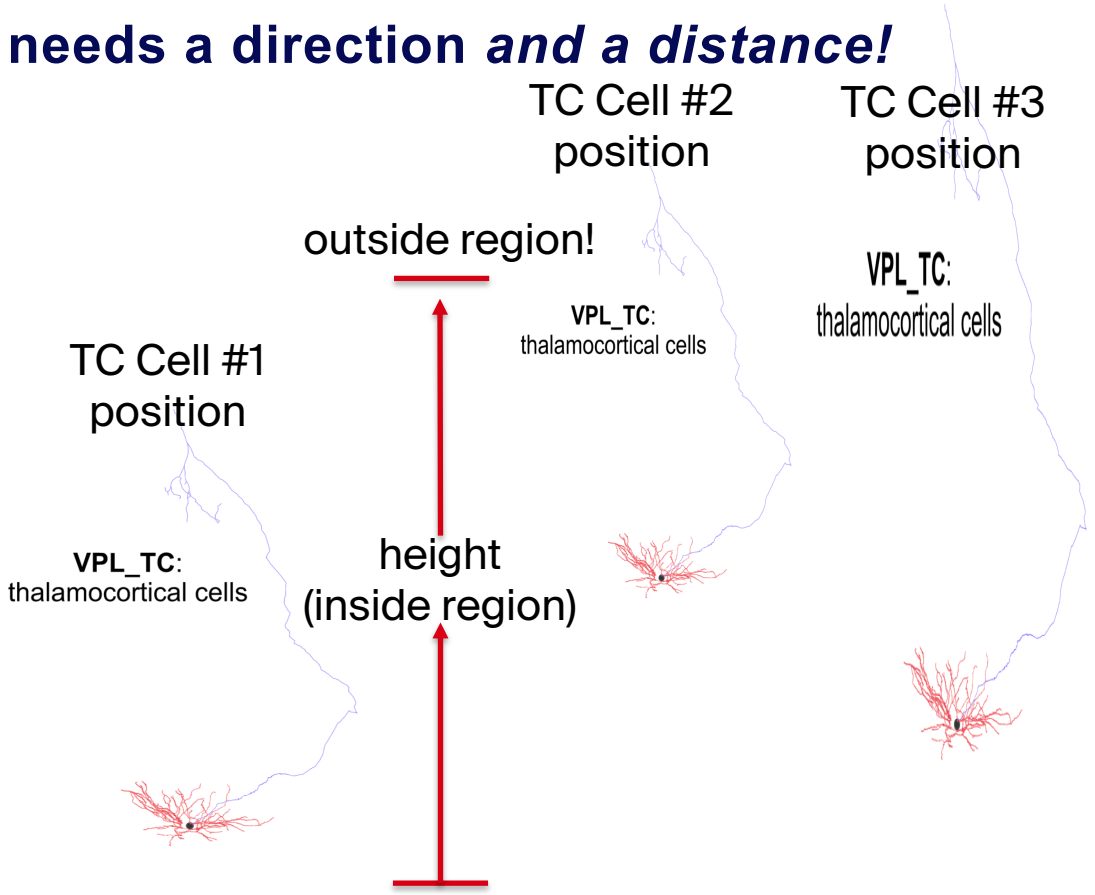
VPL_TC:
thalamocortical cells



Ref: (Iavarone, et al., 2023 https://doi.org/10.1016/j.celrep.2023.112200)

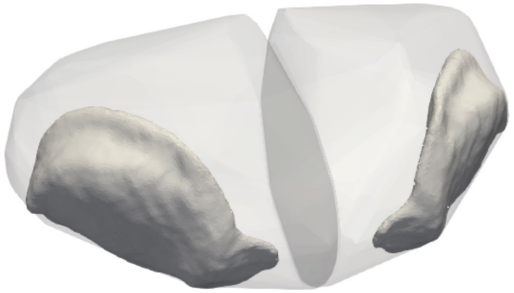
However, any good road trip needs a direction *and a distance!*





Ref: (Iavarone, et al., 2023 https://doi.org/10.1016/j.celrep.2023.112200)

We need to calculate our “distance hints” *along* each direction-vector, for every voxel

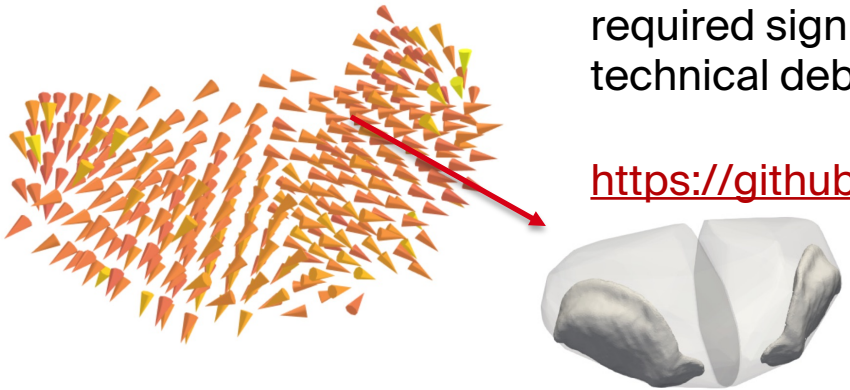


Step 1. Create 3D polygons (“meshes”) of different parts of thalamus.

Step 2. ”Draw” lines (“ray-tracing”) along each direction vector, and find the distance of that voxel to every mesh intersection.

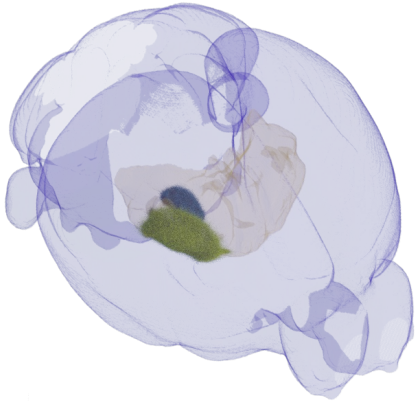
This complex technique was also created by a predecessor, but required significant rebuilding for both quality issues and much technical debt.

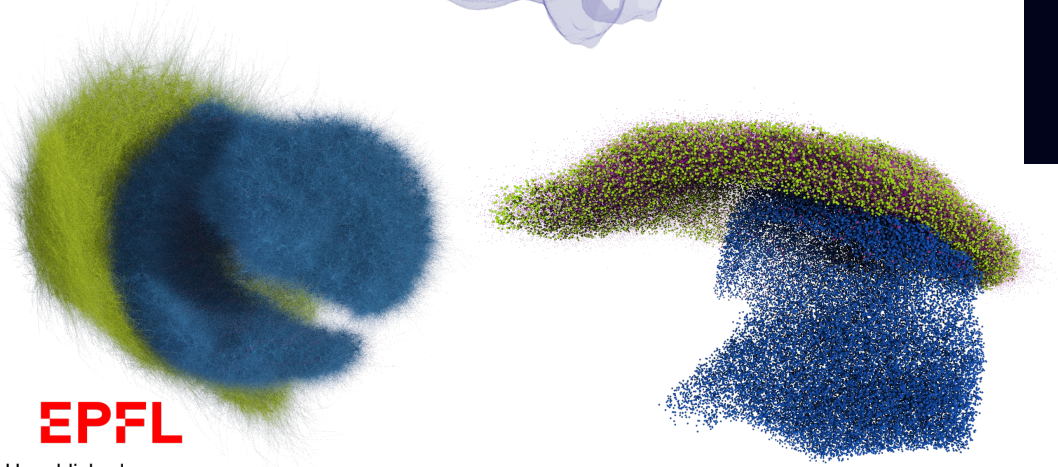
https://github.com/BlueBrain/atlas-placement-hints



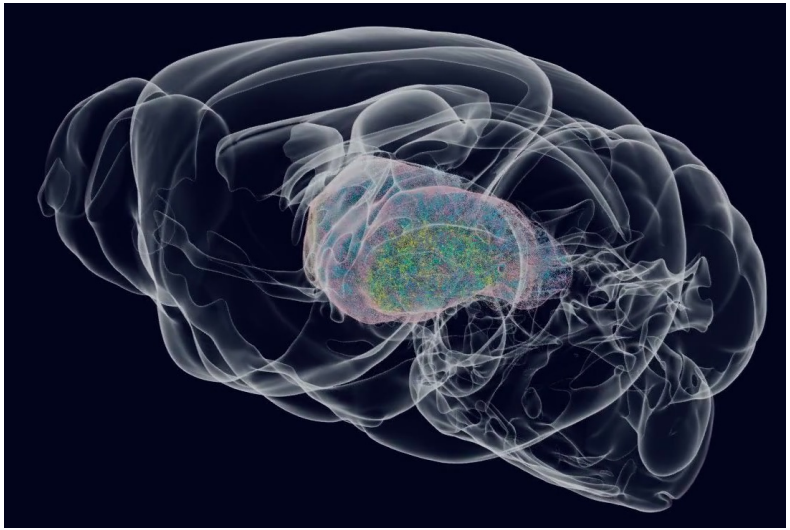
Okay…can we place the cells now??? Yes! (Reproducibly!)

Somatosensory thalamus circuit





Whole thalamus circuit

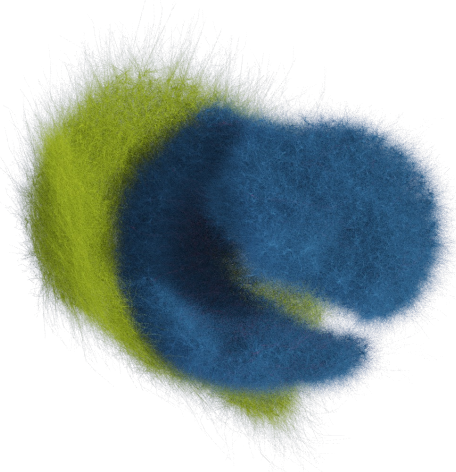


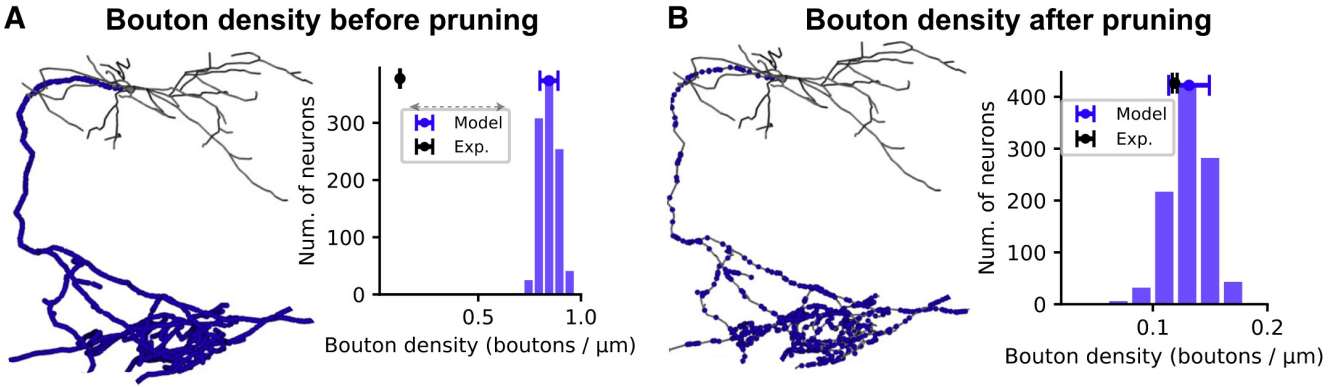
EPFL

Ref: Unpublished

However, there's a word I haven't said in a while…"synapse"

This thalamus circuit looks pretty "fuzzy"…



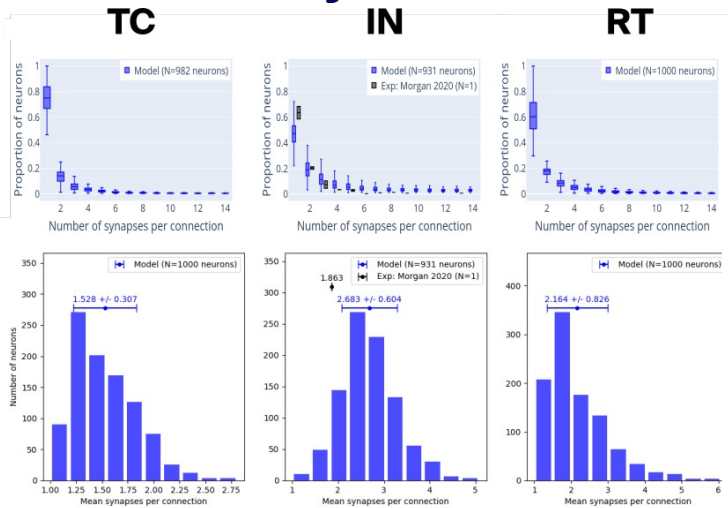


Fortunately, synapses are easier to deal with:

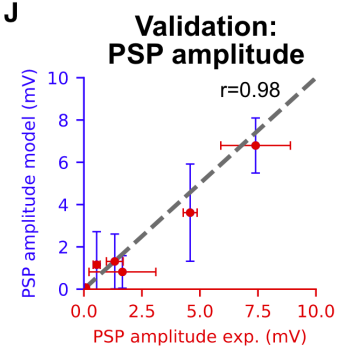
Step 1. Find the locations of **possible** synapse locations based on how close source cell axon compartments are to target cell dendrites.
Step 2. "Prune" the number of synapses down to experimentally-measured levels for that type of connection.

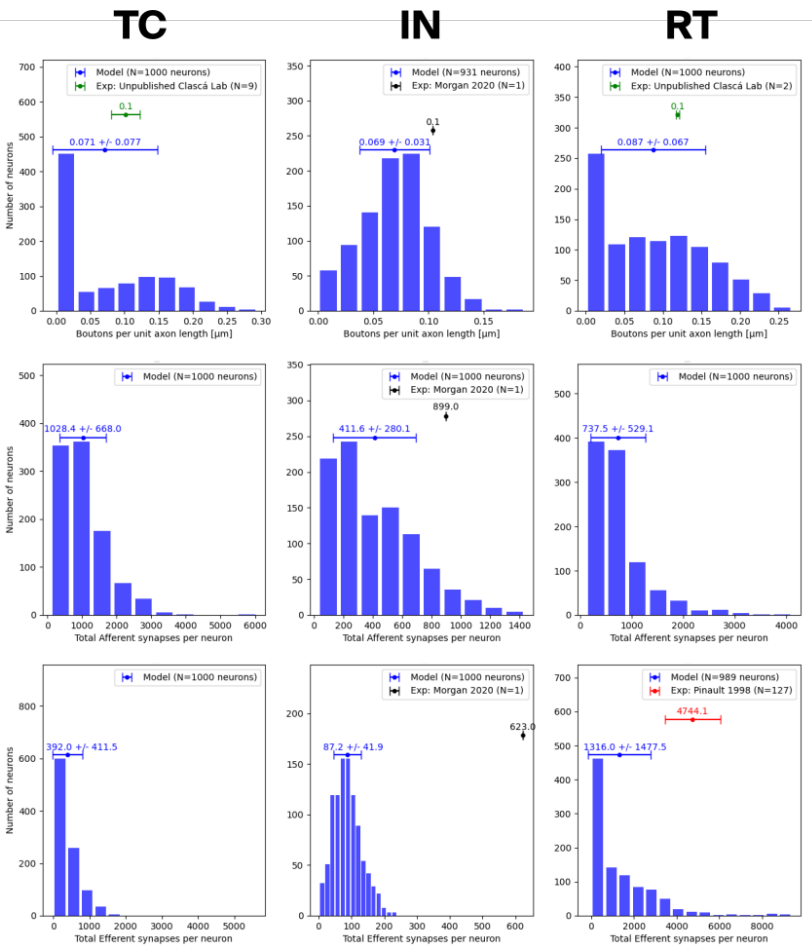
Ref: (Iavarone, et al., 2023 https://doi.org/10.1016/j.celrep.2023.112200), Unpublished

Connectivity validation



J





For validating “static” synaptic connectivity, I consolidated and parallelized a large amount of thalamus validation code spanning several years. This code can be quickly adapted to run on any SONATA circuit. (Code not public yet)

However, the dearth of precise thalamus synapse data is a big limiting factor for us (there is MUCH more for cortex!)

Ref: Unpublished, middle: (Iavarone, et al., 2023 https://doi.org/10.1016/j.celrep.2023.112200)

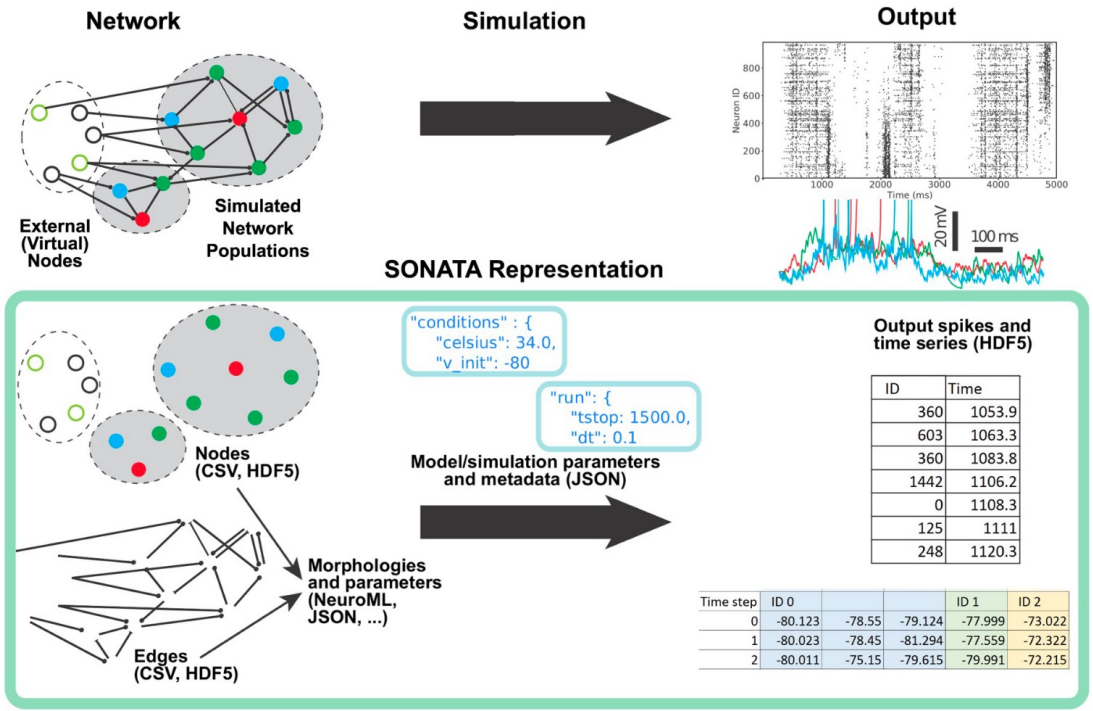
Parallelization: SONATA circuits and CoreNEURON

If y'all ever plan to run BIG simulations fast (100,000 x N cells), these will be important.

SONATA is an efficient, scalable specification for representing both your neural circuit and its output. It was co-developed between BBP and the Allen Institute and is mostly comprised of HDF5 and JSON files.

CoreNEURON is a BBP-created NEURON "engine" optimized for memory usage and speed, and has been added to NEURON as an option.

All BBP and Allen Institute circuits use SONATA format. NetPyNE has support too.



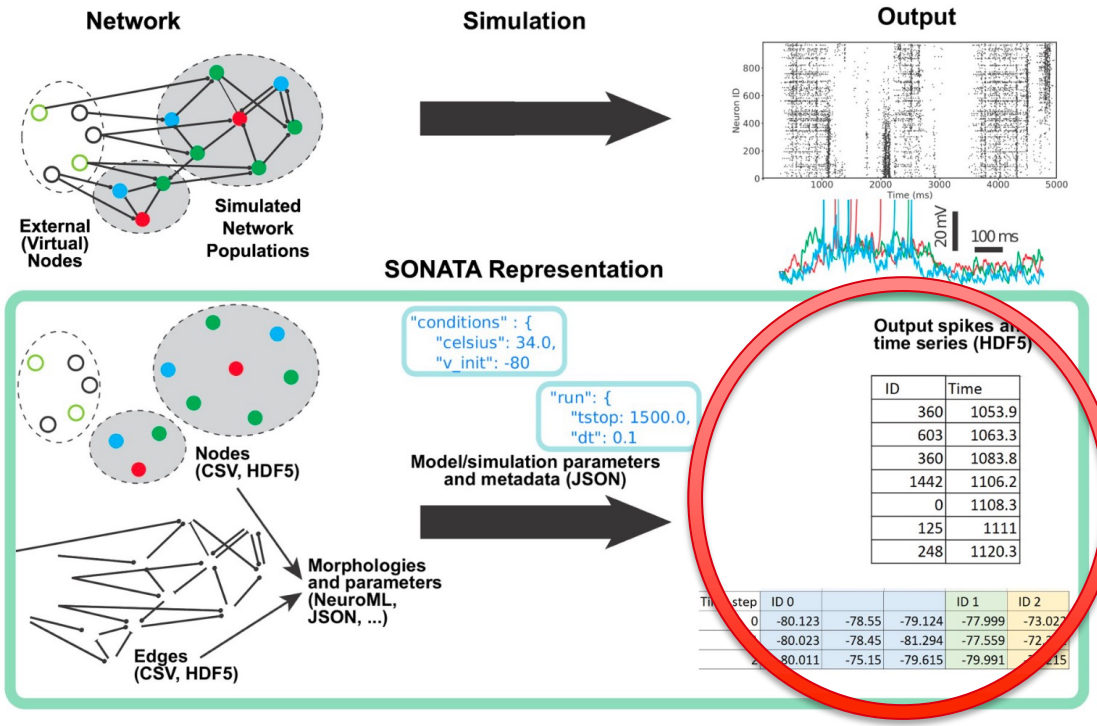
Ref: (Dai, et al., 2020 https://doi.org/10.1371/journal.pcbi.1007696)

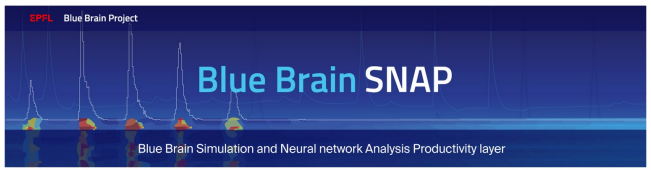
Parallelization: SONATA circuits and analysis

Raw SONATA (especially HDF5) is efficient, but not human-friendly. SNAP helps you interact with your circuit, and BlueETL runs parallel analyses on your output.

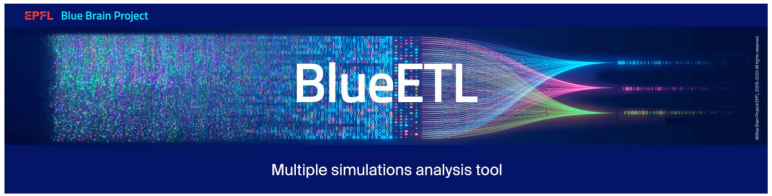
I did not help create these, but I have spent significant time learning SONATA, SNAP, and BlueETL.

BlueETL is named after the “Extract-Transform-Load” paradigm for data, and is built around Pandas’ DataFrames, which are very powerful in the right hands!





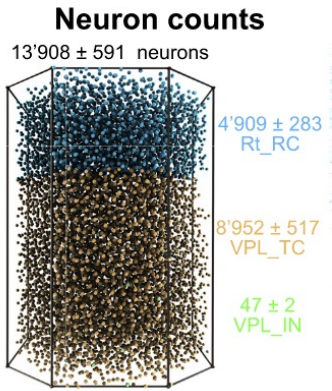
pandas

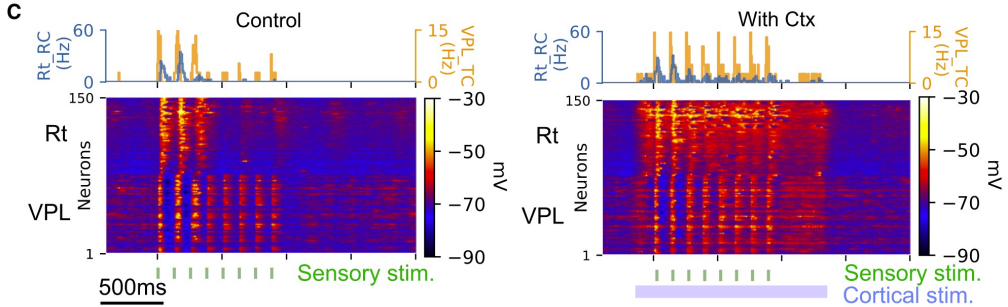


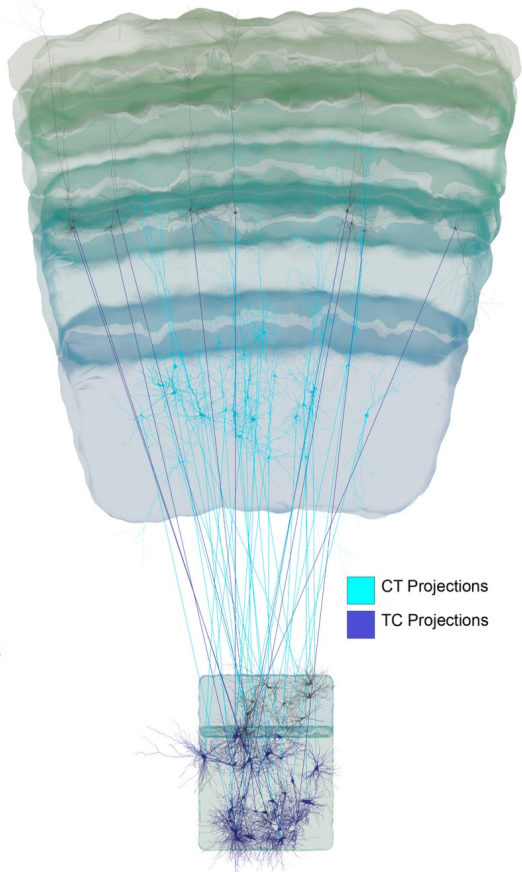
EPFL

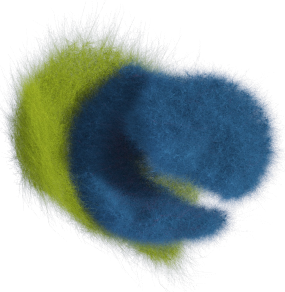
Ref: (Dai, et al., 2020 https://doi.org/10.1371/journal.pcbi.1007696)

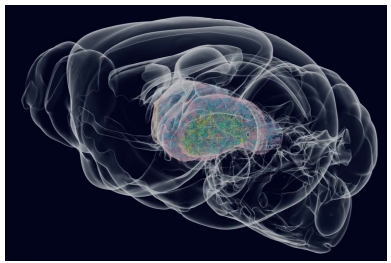
If this sounds more like “Methods” than “Results”, it is…











EPFL

Ref: Unpublished and (Iavarone, et al., 2023 https://doi.org/10.1016/j.celrep.2023.112200)

Curios

GitHub

https://github.com/asoplata

Thanks for the opportunity to come speak!

Acknowledgements:

Polina Litvak
Sean Hill
Armando Romani
Hugo Dictus
Elisabetta Iavarone
Maurizio Pezzoli
Vignan Muddapu
Mike Gevaert
Joni Herttuainen
Weina Ji

And many more...